5-2014

Handling Location Uncertainty in Event Driven Experimentation

Kartik MURALIDHARAN
Singapore Management University, kartikm.2010@smu.edu.sg

Srinivasan SESHAN
Carnegie Mellon University

Narayan RAMASUBBU
University of Pittsburgh

Rajesh Krishna BALAN
Singapore Management University, rajesh@smu.edu.sg

DOI: https://doi.org/10.1145/2611286.2611303

Handling Location Uncertainty in Event Driven Experimentation

Kartik Muralidharan
School of Information Systems
Singapore Management University
kartikm.2010@smu.edu.sg

Srinivasan Seshan
Department of Computer Science
Carnegie Mellon University
srini@cs.cmu.edu

Narayan Ramasubbu
Katz Graduate School of Business
University of Pittsburgh
nramasubbu@katz.pitt.edu

Rajesh Krishna Balan
School of Information Systems
Singapore Management University
rajesh@smu.edu.sg

ABSTRACT

The wide spread use of smart phones has ushered in a wave of context-based advertising services that operate on predefined user events. A prime example is Location Based Advertising. What is missing though, is the ability to experiment with these services under *varying* event conditions with *real* users using their regular phones in *real-world* environments. Such experiments provide greater insight into user needs for and responsiveness towards context-based advertising applications. However, these event-driven experiments rely on data that arrive from sources such as mobile sensors which have inherent uncertainties associated with them. This effects the interpretation of the outcome of an experiment. In this paper we introduce *Jarvis*, a behavioural experimentation platform that supports running in-situ real-time experiments of mobile advertising services, targeting real participants on their smart phones based on multiple context-specific events. We highlight the challenges of handling uncertainty on such a platform as well as how we deal with ambiguity in the location attribute.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Probabilistic algorithms*

Keywords

Event Processing, Context Uncertainty

1. INTRODUCTION

A Gartner report [4] stated that customers and not marketers are driving demand for context-enriched content. Context-enriched content is nothing but the information, data and other content, ranging from articles to advertising to applications, that is based on the user's context. The context here being, relevant facts about current conditions that are true in the moment but may not be in the future.

Early discussion of context-aware applications has shown the ability to use mobile sensing to infer a variety of context and build applications designed to respond in real time to changes in personal situation [8, 9, 14, 19]. However, these studies are not representative of reality as they are controlled, often with a limited set of users restricted to specific campus/office environments. To bridge this gap we are building Jarvis, a platform that facilitates a better understanding of user needs through large-scale, in-situ, real-time experimentation that require context-specific triggers.[1] The goal of Jarvis is to provide experimenters access to deeper, near-real time user context (e.g., location, activity) without the hassles of experimentation such as subject selection, bias and so on.

Of the many possibilities, a use case we envision for Jarvis, is providing retailers a platform to run *lifestyle* based experiments. For example, a coffee shop owner may want to test whether offering discount coupons to people who have been waiting outside the coffee shop for at least 10 minutes, will improve sales. However, a key challenge in running such experiments is that the trigger events are derived from context collected using built-in sensors on the mobile device. These sensors have inherent uncertainties associated with them and as a result can include people who do not satisfy the experiment criteria [3]. Continuing with the previous example, discount coupons could be sent to people who are in fact *not* outside the coffee store but are reported to be by the system as a result of localization error. It is therefore pertinent to arm experimenters with sufficient information of the possible impact of context uncertainty on the outcome of their experiment. For example, informing the experimenter that 2% of the subjects might have falsely satisfied the event conditions will assist them in defining the success criteria of

DEBS'14, May 26-29, 2014, MUMBAI, India.
Copyright 2014 ACM 978-1-4503-2737-4/14/05 ...$15.00.
http://dx.doi.org/10.1145/2611286.2611303.

[1] This work is part a research initiative that involves a large pool of opt-in participants sharing information collected on their mobile device. As of January 2014, we have 1,960 registered users.

their test. Further, defining a confidence metric for each individual subject, who satisfies the experiment requirements, provides a better understanding of the relationship between the experiment parameters. For example, if subjects considered to have a high context-confidence redeem the discount coupon, we can conclude a strong correlation between the event attributes (standing outside the shop for 10 minutes) and the content delivered. This information is important, not only for understanding user behaviour towards context-based interventions, but also towards building better context-aware systems and applications.

Providing such information unfortunately, is not trivial. The challenges are two fold: 1)Not all context generators provide the necessary information directly. Indoor localization systems for example, generally provide type II errors (false negatives) and do not measure type I errors (false positives) of the system and 2) Context uncertainty is highly dynamic and individual. For example, activity classification accuracy is dependent on the activity being classified as well the device being used. It would therefore be incorrect to have a static interpretation of error for a given context source. While techniques of increasing context confidence through redundancy or sensor fusion exist, they do not completely eliminate the need to handle context uncertainty.

In this paper, we describe the design of Jarvis, our Behavioural Experimentation Platform (BEP). More specifically we talk about the Uncertainty-Handling module within Jarvis. We show how the module defines a confidence metric for the location predicate as well as how it stochastically estimates additional information such as the number of false positives. In doing so, we provide adequate information to the experimenter to process the results of an experiment - allowing them to either either re-run the experiment (with new parameters and constraints), run a new experiment, or declare success.

2. SYSTEM OVERVIEW

In this section we give an overview of the experiment life cycle as well the Jarvis system architecture. Our architecture although not presented as a research contribution here, melds ideas from other distributed event-based systems. The goal of our system is to transform the mobile device from being merely an observer of human context to an enabler of behavioral/sociological experiments. We will also talk about how the location context information is collected as part of our system and the type of errors observed within this context source. A detailed description of our experimentation testbed is given in [15]

2.1 An Experiment Life-cycle

Figure 1 shows the sequence of steps necessary to run experiments. The sequence is:

1. A context collector application needs to be installed on participant smartphones. We currently support iOS 6+, Android 3+, and WP8+ smartphones.

2. The collector application collects sensor and context data from the phone and sends it to our real-time Event Processing Agent (EPA) where it is processed to obtain the required context triggers such as location, current activity, group status etc.

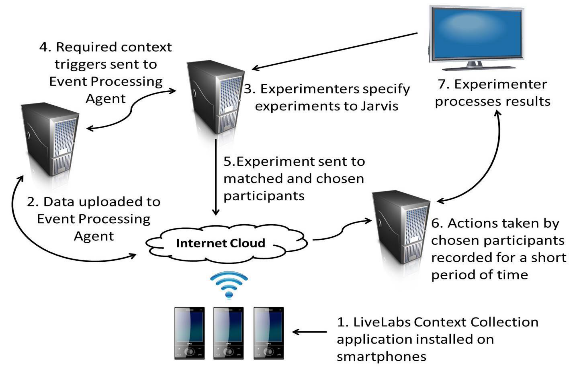


Figure 1: An Experiment life-cycle.

3. Experimenters specify their experiments using Jarvis, our Behavioural Experimentation Platform(BEP). Section 2.2 provides more details about the BEP.

4. Jarvis registers the required context triggers with the EPA. For example, "inform me when you find people standing outside the coffee shop for the past 10 minutes". The EPA server will keep track of all these events and call back the BEP when the triggers match the current context.

5. When a callback is received with the list of matched participants, Jarvis will pick a subset and send a notification with the experiment details to each selected participant. An experiment could be a discount, a request to run an application, a survey etc. Note that this subset can include participants that in reality have *not* matched all context predicates specified in the experiment but are deemed to have as a result of error in the context collected.

6. Jarvis will monitor the selected participants for a set period of time and record what they did in response to the experiment stimulus. This data is then packaged, in a privacy preserving way, and reported to the experimenter. This report also includes details of any uncertainty observed in the context. The ability to observe the entire experimental effect (both positive and negative) is a key unique property and selling point.

7. The experimenter processes the results and determines how to change their experiment (if required).

2.2 Jarvis Architecture

Figure 2 shows the various modules of the Behavioral Experimentation Platform or Jarvis. The UI, shown in Figure 3, allows experimenters to specify a wide variety of context predicates that needs to be matched by the participants. The *Context Information* module provides historical data (if available and applicable) of that context attribute. For example, if the experimenter chooses Starbucks as a target location the module displays the average population density, using a heat map, observed at that location. This information allows the experimenter to make a better judgement of the selected context attribute. Once the experiment is defined the *Experiment Validation* module ensures that the experiment is safe and valid for the participant pool. Events, such as "deliver a specific targeted discount to people at the

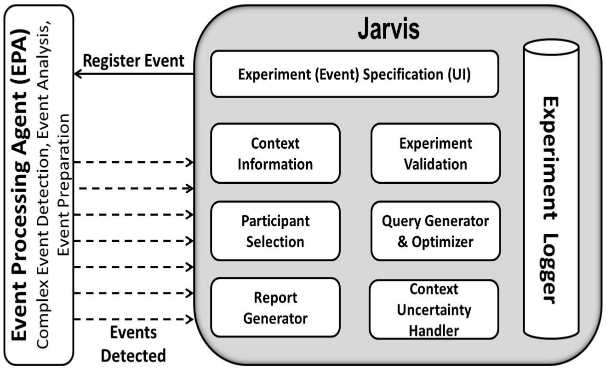


Figure 2: System Architecture

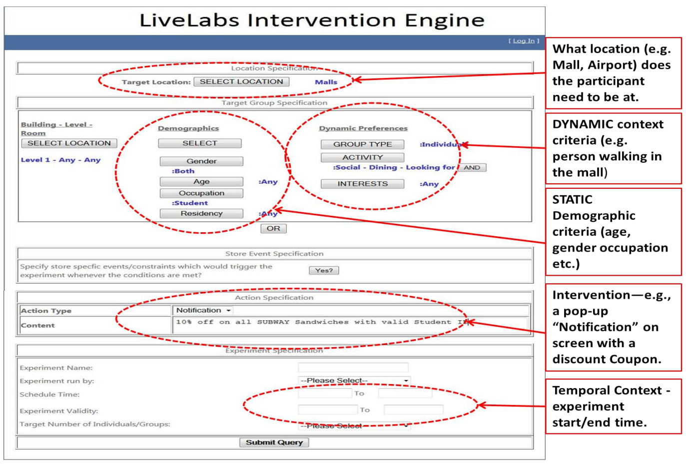


Figure 3: UI For Specifying Experiments

mall who are moving around in groups of 2", are captured using an SQL-based syntax. The context predicates are defined as a set of logical rules, that can be chained together using explicit AND operators similar to the *Where* clause of SQL. This query is generated and processed by the *Query Generator and Optimizer* module that bears many similarities to that of the Amit event processing tool [2]. When the experiment conditions are satisfied by a participant(s), the *Context Uncertainty Handler* computes the corresponding confidence metric for each of the event attributes. We use an uncertainty model based on a predicate representation of contexts and associated confidence values [16]. In Section 3 we will see how this confidence metric is computed for the location attribute. The *Participant Selection* module picks a subset of participants based on a confidence threshold specified by the experimenter. Once the participant's response to the experiment has been collected, the *Report Generator* module provides a summary of the experiment that includes an overview of the impact of the different event attributes on the experiment outcome.

2.3 Participant Location

We currently track participant location indoors at three venues - a large shopping mall, an international airport and a university campus. In order to support multiple mobile OS platforms, our localization system employs a 'reverse fingerprinting' technique by Khan [11]. In their approach, rather than relying on the Wi-Fi AP signal strength readings reported by a mobile device, they use an infrastructure-assisted solution based on querying the commercial Wi-Fi controller infrastructure.

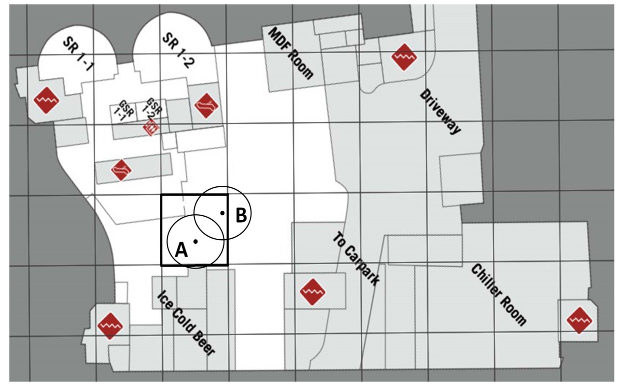


Figure 4: A floor map of our building overlayed with the location of two participants that are potential targets for a coupon from Ice Cold Beer. The inner dot represents the system detected coordinates of the participants while the outer ring represents the location error radius.

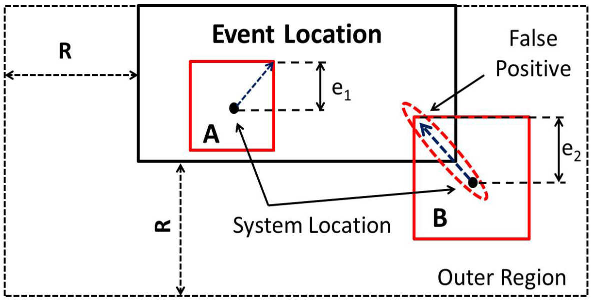
Every floor is divided into multiple zones (e.g. Shops, Classrooms) and each zone contains multiple landmarks. Identifying a participant's location means associating the participant to a landmark and in turn to a zone. Participants are considered to have satisfied the event location condition if their location is *contained in* or *touches* the zone defined in the experiment [7]. The inter-landmark distance is approximately 3 meters for the university campus and 6 meters for the shopping mall.

Using this localization technique, we observe an average location error radius of two landmarks approximately 70% of the time. There is however a caveat when computing the confidence of the location attribute. We observe that the location error distribution depends not only on the venue (and zone), but also varies with time of day and day of week. As a result, using the *static* system defined accuracy is not sufficient. Further, each individual has a certain location confidence based on their current position and error radius as reported by the system. For example in Figure 4, participant *A* reported to be at the center of a store, should have a higher probability of actually being within the experiment defined location area (even with location error) as opposed to participant *B* reported to be at the edges. It therefore becomes necessary to compute individual location confidence based on realtime observations and environment conditions. In Section 3 we will describe how our algorithm defines the location confidence for each participant as well as how it estimates the number of false positives within the set of participants that satisfy the location condition.

3. HANDLING LOCATION UNCERTAINTY

When generating a report, there are two pieces of information needed to process the outcome of an experiment. For every participant satisfying the event we need:

1. The confidence of each event attribute specified as part of an experiment.
2. The number of cases in which the event did not occur in reality (false positives).



R is the maximum error radius observed at that location. e_1 and e_2 are the respective location error radius of Participant A and Participant B such that $e_1 \leq e_2 \leq$ R .

Figure 5: Setting up the environment to compute the location confidence and the false positives of an event.

In particular when dealing with the location attribute, we want to know the location confidence of each participant as well as the number of participants that *might not* have satisfied the location requirement i.e their location is reported incorrectly to be within the event location. Note, it is not possible to infer the set of false positives based on the confidence value alone i.e. participants whose location confidence is low does not necessarily imply that the participant is originally from outside the event location. The number of false positives depends on several factors such as the area of the event location, the population density of participants within and outside the event location, the distribution of these participants as well as the location error distribution.

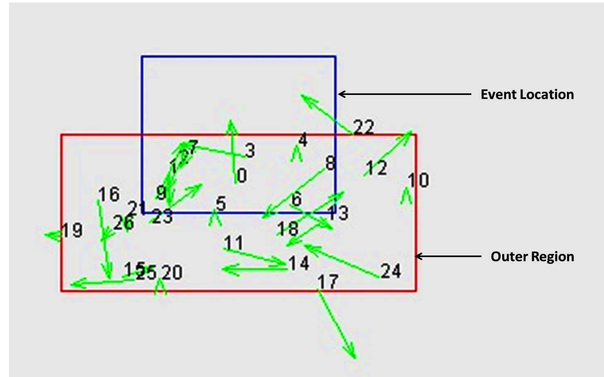
The input to our algorithm is sensor data about the location of people that *satisfy* the location condition. This data is in a (x,y) co-ordinate format, with respect to the given building, along with the error radius detected at that location. Similar to [17] all locations are expressed as minimum bounding rectangles. While approximating sensor regions with minimum bounding rectangles decreases the accuracy of location detection, the advantages in terms of performance and simplicity far outweigh the loss in accuracy. We then compute for each participant, what fraction of their minimum bounding rectangle intersects with the event location. We define this overlap ratio (ranging between 0 and 1) as the location confidence of the participant with the intuition that, larger the overlap, more likely is the participant to have satisfied the location condition specified in the experiment. Thus Participant A in Figure 5 has a location confidence of one, as the bounding rectangle is fully contained within the event location. This confidence value also serves as a first step to filter participants that do not meet the confidence requirement set by the experimenter. In Section 4 we evaluate whether this feature is a good enough metric to represent location confidence. Note that the algorithm does not compute the location confidence for Participant B in Figure 5 as it is not within the set of participants that satisfy the event location condition.

The second part of the algorithm uses Monte-Carlo methods to estimate the number of false positives - the number of participants that in reality were not within the event location. This is done through the following steps:

1. Define a region outside the event location. This is shown as a dotted line in Figure 5 surrounding the event location. The dimensions of this region is proportional to the maximum error radius R observed at that location by our indoor localization system.

2. Retrieve the system location of all participants within these two regions.

3. Apply a location error with a given error distribution, across all participants, thereby shifting the participants from their system location. As a result of this location shift, participants that were outside the event location can now be within. This is shown in Figure 5 with Participant B moving into the event location as a result of this shift. Such participants constitute the false positives of the system.



The arrow indicates the transition of a participant from their true location to the system location as a result of error. The length of the arrow represents the location error magnitude.

Figure 6: Recreating the event environment using a simulator.

We estimate the number of false positives, by emulating the test environment and observing all possible permutations of participants' location, under the given conditions. To do this, the third step is repeated 1000 times, each time shifting a participant from their system location and capturing the number of false positives as a result of this shift. For a given event location, error distribution and participant spread within and outside this region, the algorithm estimates the number of false positives as the average across these iterations. Thus if the system reports ten participants to be within the event location and our algorithm estimates two as the average number of false positives, we report that 20% of the participants were likely not within the event location. Note, as this step does not filter any participants, it is done post the experiment (e.g. after the coupon has been sent to participants and the behaviour is observed) when generating the final report for the experimenter.

4. EVALUATION

To evaluate our algorithm we require the ground truth information of each participant i.e. was the participant within the event location in reality. To get around this requirement we build a simulator to recreate our experiment environment. To do this, the simulator takes multiple inputs such as the event location dimensions, the number of participants, population density as well as the location error distribution.

Based on these inputs a set of participants are generated and placed uniformly across the simulated environment.

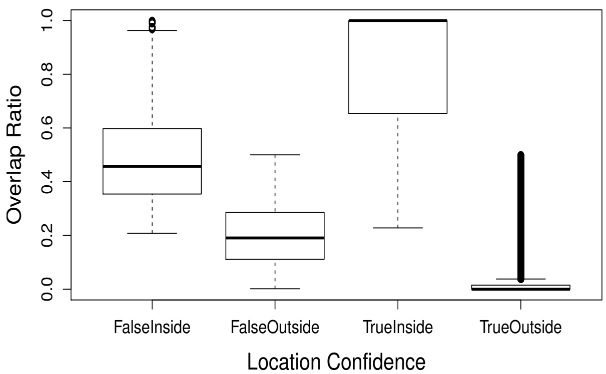


Figure 7: Box Plot capturing the distribution of overlap ratio across the four location classes of participants.

The current position of each participant constitutes the *true location* or *ground truth* of that participant. A given error distribution is then applied across all participants shifting them from their true location. The resulting position of each participant now constitutes the *system location* of that participant. Figure 6 shows a screenshot of the simulator. The arrow captures the shift of each participant from their true location to their system defined location. The length of the arrow represents the magnitude of the location error. Given the (true location, system location) pair, we can evaluate the accuracy of our algorithm in estimating the number of false positives in a given event as well as measuring the reliability of using overlap ratio to represent location confidence.

4.1 Using overlap ratio to represent location confidence

To evaluate the accuracy of using overlap ratio as our location confidence metric, we divide the participants into four classes: 1) *TrueInside*-Participants whose true location and system location coordinates are inside the event location, 2) *TrueOutside*-Participants whose true location and system location coordinates are outside the event location, 3) *FalseInside*-Participants whose true location is outside the event location but the system location is within the region and 4) *FalseOutside*-Participants whose true location is within the event location but the system location is outside the region. Note, we are truly only interested in two classes of participants, *TrueInside* and *FalseInside*, as these are the set of participants considered to have satisfied the event location condition. We however include all four classes in our observation of whether overlap ratio - area of intersection with the event location by the area of the minimum bounding rectangle as defined by the system location- is a good indicator of location confidence.

The overlap ratio was captured for ten uniform error distribution scenarios with the maximum error radius R, ranging from 3 to 12 meters for each scenario. Each scenario was repeated 100 times, randomly generating the number of participants (min. 20, max. 100) during each run. The event location dimensions remained constant for the complete experiment, while the dimensions of the region outside the event location varied based on the maximum error radius for the given location error distribution scenario.

Figure 7 captures the distribution of the overlap ratio across all four location classes. We observe that the box plot for each class does not overlap significantly, suggesting the use of the overlap ratio to differentiate between the different classes of participants. Thus participants with a higher overlap ratio are more likely to be within the event location than participants with a lower ratio. We further evaluate it's classification capability by building a Naive Bayes model in Weka using overlap ratio as the feature . The resulting model provides an accuracy of 84.6% in classifying a participant's true location based on the overlap ratio. Unfortunately, classification errors do still exist. Of interest are those participants classified as *TrueInside* when in reality they should be classified as *FalseInside* - which is the reason why we attempt to estimate the number of false positives. However, despite these errors, we still consider overlap ratio to be a good representation of location confidence. Table 1 summarizes the mean and standard deviation of each class observed during the simulation run.

Class	Mean	SD	SE
FalseInside	0.4848	0.1724	0.0032
FalseOutside	0.2049	0.1177	0.0019
TrueInside	0.8315	0.2171	0.0027
TrueOutside	0.0391	0.0857	0.0004

All differences are significant (using student's t-test with p = 0.05).

Table 1: Mean and Standard Deviation of overlap ratio across the different location classes.

P(N)	P(E)	FP(True)	FP(Estimate)	Estimate Error (%)
20	8	0	1	12.5

Where *P(N)* is the total number of participants during the simulation run, *P(E)* is the number of participants that satisfied the Event Location condition and *FP()* is the number of false positives.

Table 2: Simulator Output for a single run with maximum error radius *R=3m*.

4.2 Accuracy in estimating the number of False Positives

Given that the simulator captures both the true location and system location of each participant, we can compare the *true* number of false positives with the value *estimated* by our algorithm. We run the simulator for ten different uniform location error distribution scenarios (R=3 to 12 meters), with each scenario repeated 100 times. The number of participants (min. 20, max. 100) were randomly generated during each run. For each run, the number of false positives was then computed using the algorithm described in Section 3. Table 2 captures the output of the simulator for a single run.

Figure 8 shows a cdf plot of the percentage error in estimating the number of false positives by our algorithm. We observe that 80% of the time our algorithm can accurately estimate the number of false positives in an event with an error less than ±25%.

4.3 Future Work

An important goal of our system is to provide a platform that facilitates getting a better understanding of user be-

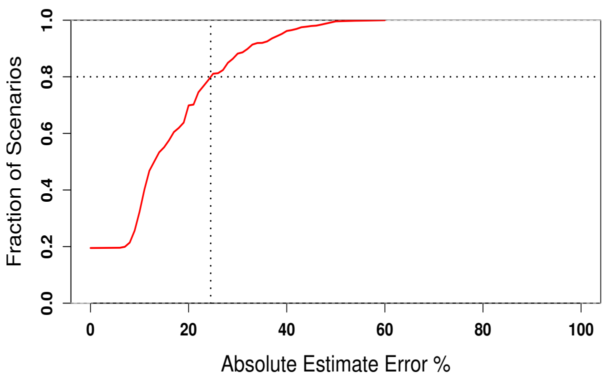


Figure 8: CDF of the % error in estimating the number of participants that did not satisfy the event conditions (false positives).

haviour towards context-based systems, through a process of experimentation. To answer questions such as "Does sending a coupon to consumers standing outside a coffee shop improve sales?" would require sending coupons to participants outside a coffee shop as well as other locations. In order to validate the outcome of that experiment, knowing the location confidence of the participants involved as well as the confidence of other event attributes are important. A high location confidence among the set of participants that adopt the coupon can suggest a strong correlation between coupon adoption and location. It therefore becomes pertinent to provide such confidence information to experimenters. However, instead of providing raw confidence metrics a report that includes a statistical analysis of these values will perhaps be more readable by an experimenter. We are currently exploring additional information that should go into such a report that will be relevant and useful to the users of our platform.

While reporting the confidence of event attributes is a key aspect of our system we are also investigating other sources of input that can help reduce this uncertainty. One such input is explicit user feedback [6]. Asking users whether their current context matches the event attributes defined by the experiment will eliminate any ambiguity. However, in order for such a system to prove useful, we require significant feedback. There is a need to therefore identify the right set of incentives as well as explore the right balance of questions given the smartphone screen-estate and user effort required.

5. RELATED WORK

Context-aware systems can't always identify the current context precisely, hence they need support for handling uncertainty. Various mechanisms such as probabilistic logic, fuzzy logic and Bayesian networks are used for reasoning about uncertainty [5, 10, 17]. MiddleWhere [17] uses probabilistic reasoning techniques to deduce a person's location confidence. However, this technique assumes the availability of precise information associated with the location sensing technology, such as the probability of a false positive. In contrast, we assume no such information is readily available and instead attempt to estimate the number of false positives for a given scenario.

There is also considerable work in reducing context uncertainty using sensor fusion [13, 18] as well as through user mediation [6, 9]. These efforts are orthogonal to our work, where we focus on representing location uncertainty, as opposed to reducing it, and associating this uncertainty with the outcome of an event.

Finally, while uncertainty is a significant problem for many other ubiquitous computing applications, it is not *as* problematic for advertising. From the advertiser's perspective, any reduction in uncertainty is welcome. Taking this viewpoint, current context based advertising applications (commercial and research driven) [1, 12, 19] do not handle or represent uncertainty in their context stream. As a result, any visibility as to why a consumer did not react to a given stimulus is ignored - something which our platform intends on correcting.

6. CONCLUSION

In this paper we outline the design of a Behavioural Experimentation Platform, that attempts to gain insight into consumer behaviour towards context based advertising, through event based experimentation. As these experiments rely on uncertain context, there is a need to identify as well as quantify this uncertainty. We describe how our platform defines a confidence metric for the location attribute as well as how it derives information, such as the number of false positives. Our evaluation shows using overlap ratio to represent location confidence is reliable and that our algorithm to estimate the number of false positives has minimal errors. Both these values are important in understanding the outcome of an experiment and in turn defining it's success criteria.

7. ACKNOWLEDGMENTS

This research is supported by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive Digital Media Programme Office, Media Development Authority and supported in part by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

8. REFERENCES

[1] *Shopkick*. `http://www.shopkick.com/`.
[2] A. Adi and O. Etzion. Amit - the situation manager. *The VLDB Journal*, 13(2):177–203, May 2004.
[3] A. Artikis, O. Etzion, Z. Feldman, and F. Fournier. Event processing under uncertainty. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 32–43, New York, NY, USA, 2012. ACM.
[4] P. Basiliere. Customer communications management software providers must meet the demands of the nexus of forces. Technical report, Gartner Report, Nov 2012.
[5] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz. A probabilistic room location service for wireless networked environments. In *Proceedings of the 3rd International Conference on Ubiquitous Computing*, UbiComp '01, pages 18–34, London, UK, UK, 2001. Springer-Verlag.
[6] A. Dey, J. Mankoff, G. Abowd, and S. Carter. Distributed mediation of ambiguous context in aware

environments. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 121–130. ACM, 2002.

[7] O. Etzion and P. Niblett. *Event Processing in Action*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2010.

[8] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10. ACM, 2010.

[9] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay. Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys '07. ACM, 2007.

[10] D. Guan, W. Yuan, A. Gavrilov, S. Lee, Y. Lee, and S. Han. Using fuzzy decision tree to handle uncertainty in context deduction. In D.-S. Huang, K. Li, and G. Irwin, editors, *Computational Intelligence*, volume 4114 of *Lecture Notes in Computer Science*, pages 63–72. Springer Berlin Heidelberg, 2006.

[11] A. Khan, V. Ranjan, T.-T. Luong, R. Balan, and A. Misra. Experiences with performance tradeoffs in practical, continuous indoor localization. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–9, June 2013.

[12] B. Kim, J.-Y. Ha, S. Lee, S. Kang, Y. Lee, Y. Rhee, L. Nachman, and J. Song. Adnext: A visit-pattern-aware mobile advertising system for urban commercial complexes. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, HotMobile '11, pages 7–12, New York, NY, USA, 2011. ACM.

[13] I. König, B. N. Klein, and K. David. On the stability of context prediction. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp '13 Adjunct, pages 471–480, New York, NY, USA, 2013. ACM.

[14] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song. Comon: cooperative ambience monitoring platform with continuity and benefit awareness. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, MobiSys '12. ACM, 2012.

[15] A. Misra and R. K. Balan. Livelabs: Initial reflections on building a large-scale mobile behavioral experimentation testbed. *SIGMOBILE Mob. Comput. Commun. Rev.*, 17(4):47–59, Dec. 2013.

[16] A. Ranganathan, J. Al-Muhtadi, and R. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *Pervasive Computing, IEEE*, 3(2):62–70, April 2004.

[17] A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. Campbell, and M. D. Mickunas. MiddleWhere: A middleware for location awareness in ubiquitous computing applications. In *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware*, Middleware '04, pages 397–416, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[18] S. Sigg, S. Haseloff, and K. David. An alignment approach for context prediction tasks in ubicomp environments. *Pervasive Computing, IEEE*, 9(4):90–97, October 2010.

[19] B. Yan and G. Chen. Appjoy: personalized mobile application discovery. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys '11. ACM, 2011.